4. (Original) The device of claim 3 wherein the device is configured to perform as part of its crypto-search operation a determination of whether a pattern match exists between a search key that is representative of data desired to be retrieved from the mass storage medium and a data signal that is representative of the decrypted data stream.

5. (Original) The device of claim 2 wherein the programmable logic device is configured to (1) receive a stream of encrypted compressed data from the mass storage medium, (2) decrypt the received stream to create a decrypted compressed data stream, (3) decompress the decrypted compressed data stream to create a decompressed decrypted data stream, and (4) perform a search operation within the decompressed decrypted data stream.

6. (Original) The device of claim 5 wherein the search operation comprises determining whether a pattern match exists between a search key that is representative of data desired to be retrieved from the mass storage medium and a data signal that is representative of the decompressed decrypted data stream.

7. (Original) The device of claim 2 wherein the programmable logic device is an FPGA.

8. (Original) The device of claim 1 wherein said data manipulation includes at least a compression operation.

9. (Original) The device of claim 1 wherein said data manipulation includes at least a decompression operation.

10. (Original) The device of claim 1 wherein said data manipulation includes at least a data reduction operation.

11. (Original) The device of claim 1 wherein said data manipulation includes at least a data classification operation.

12. (Original) The device of claim 1 wherein the device interfaces the mass storage medium with a system bus, and wherein a computer system is configured to access the system bus to communicate data processing requests to the device.

13. (Original) The device of claim 1 wherein the device is in communication with the mass storage medium over a computer network.

14. (Original) The device of claim 13 wherein the computer network is the Internet.

15. (Currently Amended) The device of claim 1 wherein the device interfaces the mass storage medium with a system bus, wherein a computer system is configured to access the system bus over a computer network to communicate data processing requests to the device~~, and wherein the device is in communication with the mass storage medium over a computer network~~.

16. (Original) The device of claim 1 wherein the mass storage medium comprises a disk system having a plurality of disks on which data is stored and a plurality of heads for reading data from the disks, wherein the programmable logic device is configured to (1) receive a plurality of continuous data streams from the mass storage medium, each data stream being received from a different head, and (2) in parallel, perform the plurality of processing operations on each received continuous data stream.

17. (Original) A method of manipulating data moving to or from a mass storage medium in a continuous stream, the method comprising:

receiving a continuous data stream moving to or from a mass storage medium; and

manipulating data in said continuous stream with reconfigurable hardware logic.

18. (Original) The method of claim 17 wherein said reconfigurable hardware logic is implemented on an FPGA.

19. (Original) The method of claim 18 wherein said manipulating step comprises:

decrypting an encrypted data stream to create a decrypted data stream; and

searching the decrypted data stream for the presence of a search key therein.

20. (Original) The method of claim 19 wherein the search key is representative of data sought to be retrieved, and wherein the searching step comprises searching the decrypted data stream by framelessly comparing and correlating the search key with a data signal representative of the decrypted data stream.

21. (Original) The method of claim 18 wherein said manipulating step comprises:

decrypting an encrypted compressed data stream to create a decrypted compressed data stream;

decompressing the compressed data stream to create a decompressed decrypted data stream; and

searching the decompressed decrypted data stream for the presence of a search key therein.

22. (Original) The method of claim 21 wherein the search key is representative of data sought to be retrieved, and wherein the searching step comprising searching the decompressed decrypted data stream by framelessly comparing and correlating the search key with a data signal representative of the decompressed decrypted data stream.

23. (Original) The method of claim 18 wherein said manipulating step includes performing a search operation.

24. (Original) The method of claim 18 wherein said manipulating step includes performing a compression operation.

25. (Original) The method of claim 18 wherein said manipulating step includes performing a decompression operation.

26. (Original) The method of claim 18 wherein said manipulating step includes performing a data reduction operation.

27. (Original) The method of claim 18 wherein said manipulating step includes performing a data classification operation.

28. (Original) The method of claim 18 wherein a template loaded onto the re-configurable logic device defines the manipulating step, the method further comprising:

storing a plurality of templates, each defining a different manipulation operation; and

selecting a stored template for loading onto the FPGA.

29. (Currently Amended) For a programmable logic device in communication with a mass storage medium, the programmable logic device being configured to process data moving to or from the mass storage medium in accordance with a template loaded thereon, the template defining one or more processing functions, each function having an associated performance characteristic for data processing performed thereby, a method for selecting a template for programming the programmable logic device, the method comprising:

selecting a stored template from a plurality of stored templates for loading into the programmable logic device at least partially on the basis of the determined performance characteristics for each function defined by the templates, the plurality of stored templates each having a type associated with the data processing performed thereby; and

replacing a template on the programmable logic device with the selected template, the type of the selected template being the same as the type of the template on the programmable logic device.

30. (Original) The method of claim 29 wherein the at least one performance characteristic comprises data throughput.

31. (Original) The method of claim 29 wherein the at least one performance characteristic comprises an amount of programmable logic device resources consumed by the function.

32. (Original) The method of claim 29 wherein the selecting step comprises selecting the stored templates based at least in part on at least two performance characteristics for the functions, the at least two performance characteristics comprising data throughput and amounts of programmable logic device resources consumed by each function.

33. (Original) The method of claim 32 wherein the functions comprise at least one selected from the group consisting of a search operation, a data reduction operation, a data classification operation, an encryption operation, a decryption operation, a compression operation, and a decompression operation.

34. (Original) The method of claim 33 wherein the functions comprise at least one selected from the group consisting of encryption and decryption.

35. (Original) The method of claim 32 wherein the programmable logic device is an FPGA, and wherein the selecting step comprises selecting the template according to a predetermined algorithm based on the determined data throughput values and resources values for the functions defined by each template.

36. (Original) The method of claim 35 wherein the selecting step further comprises selecting the stored template that maximizes data throughput under a constraint that the functions of the selected template cannot require resources on the FPGA that exceed an amount of available resources on the FPGA.

37. (Original) The method of claim [[34]] 35 wherein the selecting step further comprises selecting the stored template that minimizes resources under a constraint that the functions of the selected template cannot have a determined data throughput value less than a pre-determined threshold value.

38. (Original) The method of claim 29 wherein the selecting step is performed dynamically as the programmable logic device receives a request to retrieve and process data stored in the mass storage medium.

39. (Canceled)

40. (Currently Amended) ~~The system of claim 39~~ A data processing system comprising:

a data storage medium;

a processing device in communication with the data storage medium; and a computer system having a system bus, wherein the computer system is configured to communicate with the processing device over the system bus;

wherein the processing device comprises a programmable logic device configured to process the data, as it passes between the data storage medium and the computer system, through a plurality of stages implemented on the programmable logic device as a processing pipeline, each processing stage being dedicated to a different processing operation; and

wherein the processing operations comprise at least two selected from the group consisting of a search operation, a data reduction operation, a data classification operation, an encryption operation, a decryption operation, a compression operation, and a decompression operation.

41. (Original) The system of claim 40 wherein one of the at least two processing operations is a search operation.

42. (Original) The system of claim 41 wherein the data storage medium comprises data stored therein in an encrypted format, and wherein the programmable logic device is further configured to (1) receive a continuous stream of encrypted data from the data storage medium, (2) decrypt the received continuous stream to create a decrypted data stream, and (3) perform a search operation within the decrypted data stream.

43. (Original) The system of claim 42 wherein the search operation is configured to determine whether a pattern match exists between a search key that is representative of data desired to be retrieved from the data storage medium and a data signal that is representative of the decrypted data stream.

44. (Original) The system of claim 41 wherein the data storage medium comprises data stored therein in an encrypted compressed format, and wherein the programmable logic device is further configured to (1) receive a stream of encrypted compressed data from the data storage medium, (2) decrypt the received stream to create a decrypted compressed data stream, (3) decompress the decrypted compressed data stream to create a decompressed decrypted data stream, and (4) perform a search operation within the decompressed decrypted data stream.

45. (Original) The system of claim 44 wherein the search operation is configured to determine whether a pattern match exists between a search key that is representative of data desired to be retrieved from the data storage medium and a data signal that is representative of the decompressed decrypted data stream.

46. (Original) The system of claim 41 wherein the programmable logic device is an FPGA.

47. (Original) The system of claim 40 wherein one of the at least two processing operations is a compression operation.

48. (Original) The system of claim 40 wherein one of the at least two processing operations is a decompression operation.

49. (Original) The system of claim 40 wherein one of the at least two processing operations is a data reduction operation.

50. (Original) The system of claim 40 wherein one of the at least two processing operations is a data classification operation.

51. (Currently Amended) The system of claim [[39]] 40 wherein the data storage medium comprises a disk drive system for magnetically storing data, the disk drive system comprising:

a rotatable disk upon which data is magnetically stored in a plurality of discontiguous arcs, wherein each arc possesses a substantially constant curvature, the plurality of discontiguous arcs together defining a generally helical pattern about a central origin;

a device for rotating the disk when data is to be read therefrom;

a read head positioned for reading the data stored on the disk as the disk rotates; and

a positioning system configured to position the read head over the disk such that, as the disk rotates, the read head follows the generally helical pattern of the discontiguous arcs.

52. (Currently Amended) The system of claim [[39]] 40 wherein a plurality of data files are stored in the data storage medium, each data file being stored as a sequence of segments, each segment having a size that is a power of 2.

53. (Canceled)

54. (Currently Amended) ~~The accelerator of claim 53~~ A hard disk drive accelerator for connection between a hard disk drive and a processor, said accelerator comprising reconfigurable hardware logic arranged such that data read from the hard disk drive streams through the reconfigurable hardware logic prior to being passed on to the processor, wherein the reconfigurable hardware logic is configured to process the data stream through pipeline comprising a plurality of processing stages, each processing stage being configured to perform a data processing operation on the data it receives, and wherein the processing operations performed by the stages of the pipeline are [[any]] at least two selected from the group consisting of: a search operation, a data reduction operation, a data classification operation, an encryption operation, a decryption operation, a compression operation, and a decompression operation.

55. (Original) The accelerator of claim 54 wherein the reconfigurable hardware logic is implemented on a programmable logic device, wherein the hard disk drive comprises data stored therein in an encrypted format, and wherein the programmable logic device is configured to (1) receive a continuous stream of encrypted data from the hard disk drive, (2) decrypt the received continuous stream to create a decrypted data stream, and (3) perform a search operation within the decrypted data stream.

56. (Original) The accelerator of claim 55 wherein the search operation is configured to determine whether a pattern match exists between a search key that is representative of data desired to be retrieved from the hard disk drive and a data signal that is representative of the decrypted data stream.

57. (Original) The accelerator of claim 54 wherein the reconfigurable hardware logic is implemented on a programmable logic device, wherein the hard disk drive comprises data stored therein in an encrypted compressed format, and wherein the programmable logic device

is configured to (1) receive a stream of encrypted compressed data from the hard disk drive, (2) decrypt the received stream to create a decrypted compressed data stream, (3) decompress the decrypted compressed data stream to create a decompressed decrypted data stream, and (4) perform a search operation within the decompressed decrypted data stream.

58. (Original) The accelerator of claim 57 wherein the search operation is configured to determine whether a pattern match exists between a search key that is representative of data desired to be retrieved from the mass storage medium and a data signal that is representative of the decompressed decrypted data stream.

59. (Original) The accelerator of claim 54 wherein the re-configurable hardware logic is implemented on an FPGA.

60. (Original) The accelerator of claim 54 wherein the processing operation of at least one stage is a search operation.

61. (Original) The accelerator of claim 54 wherein the processing operation of at least one stage is a compression operation.

62. (Original) The accelerator of claim 54 wherein the processing operation of at least one stage is a decompression operation.

63. (Original) The accelerator of claim 54 wherein the processing operation of at least one stage is a data reduction operation.

64. (Original) The accelerator of claim 54 wherein the processing operation of at least one stage is a data classification operation.

65. (Currently Amended) A device for compressing data, the device comprising:

a programmable logic device in communication with a data storage medium, the programmable logic device being configured to (1) receive data from a data source, (2) perform an encryption operation on the received data to thereby create encrypted data, (3) perform a compression operation on the ~~received~~ encrypted data to thereby create compressed data, and ~~(3)~~ (4) store the compressed data in the data storage medium.

66. (Original) The device of claim 65 wherein the compression operation is a lossless compression operation.

67. (Original) The device of claim 66 wherein the lossless compression operation is LZ compression.

68. (Original) The device of claim 65 wherein the programmable logic device is an FPGA, and wherein the data source is a computer system in communication with the FPGA via a bus.

69. (Currently Amended) A device for decompressing data, the device comprising:

a programmable logic device in communication with a data storage medium, the data storage medium comprising data stored therein in a compressed format, the programmable logic device being configured to (1) receive a continuous stream of compressed data from the data storage medium, ~~and~~ (2) perform a decompression operation on the received continuous stream of compressed data to thereby create decompressed data, and (3) perform a decryption operation on the decompressed data.

70. (Original) The device of claim 69 wherein the decompression operation is a lossless decompression operation.

71. (Original) The device of claim 70 wherein the lossless decompression operation is LZ decompression.

72. (Original) The device of claim 69 wherein the programmable logic device is an FPGA, and wherein the FPGA is further configured to perform a search operation on the decompressed data.

73. (Original) A data storage medium upon which data is stored magnetically for subsequent retrieval by a magnetic read head, the medium comprising:

a rotatable magnetic medium; and

a plurality of discontiguous arcs located on the magnetic medium for storing data.

74. (Original) The medium of claim 73 wherein the plurality of discontiguous arcs together define a generally helical pattern on the magnetic medium about a central origin.

75. (Original) The medium of claim 74 wherein each discontiguous arc possesses a substantially constant curvature.

76 (Original) The medium of claim 75 wherein the magnetic medium is a disk on which digital data is magnetically stored.

77. (Original) The medium of claim 76 wherein the plurality of discontiguous arcs together define a generally helical pattern such that, beginning from a discontiguous arc positioned at a shortest radial distance from the central origin, each successive discontiguous arc along the generally helical pattern is positioned at a radial distance from the central origin that is greater than the radial distance for the previous discontiguous arc along the generally helical pattern, and wherein each discontiguous arc spans an angle of $2\pi/W$ from the central origin, wherein W represents a total number of discontiguous arcs encountered by a read head during a single revolution of the rotatable magnetic medium.

78. (Original) The medium of claim 76 wherein a uniform distance radially separates each successive discontiguous arc along the generally helical pattern.

79. (Original) The medium of claim 76 wherein the disk is a hard disk contained within a hard disk drive.

80. (Original) The medium of claim 76 wherein each discontiguous arc includes a servo pattern recorded thereon.

81. (Original) A method of reading data from a rotatable planar magnetic storage medium upon which data is stored on a plurality of discontiguous circular arcs, and wherein

the plurality of discontiguous circular arcs together define a generally helical pattern about a central origin, the method comprising:

rotating the magnetic storage medium; and

positioning a read head to follow the generally helical pattern on the magnetic storage medium as the storage medium rotates.

82. (Original) The method of claim 81 wherein each discontiguous circular arc includes a servo pattern thereon, and wherein the positioning step comprises positioning the read head based at least in part upon a sensing of the discontiguous circular arcs' servo patterns.

83. (Original) A disk drive system for magnetically storing data, the system comprising:

a rotatable disk upon which data is magnetically stored in a plurality of discontiguous arcs, wherein each arc possesses a substantially constant curvature;

a device for rotating the disk when data is to be read therefrom;

a read head positioned for reading the data stored on the disk as the disk rotates; and

a positioning system configured to position the read head over the disk such that, as the disk rotates, the read head follows the generally helical pattern of the discontiguous arcs.

84. (Original) The system of claim 83 wherein the plurality of discontiguous arcs together define a generally helical pattern about a central origin.

85. (Original) The system of claim 84 wherein the plurality of discontiguous arcs together define a generally helical pattern such that, beginning from a discontiguous arc positioned at a shortest radial distance from the central origin, each successive discontiguous arc along the generally helical pattern is positioned at a radial distance from the central origin that is greater than the radial distance for the previous discontiguous arc along the generally helical pattern, and wherein each discontiguous arc spans an angle of 2π/W from the central origin, wherein W represents a total number of discontiguous arcs encountered by a read head during a single revolution of the rotatable magnetic medium.

86. (Currently Amended) A method of storing a data file on a storage medium, the data file having a file size comprising a total number of bytes therein, the method comprising:

determining whether the file size is an even power of 2;

if the file size is an even power of 2, requesting a block of storage space on the storage medium equal to the file size;

if the file size is not an even power of 2, requesting a plurality of blocks of storage space on the storage medium, each block having a size that is equal to a power of 2; and

if the request is accepted, storing the data file on the storage medium as one or more data file segments in accordance with the request.

87. (Original) The method of claim 86 wherein the file size can be represented in binary as $F = F_k \ldots F_2F_1$, and wherein if the file size is not an even power of 2, the requesting step comprises requesting a total number n of blocks $B_1, \ldots, B_n$ equal to a total number of bits in F equal to 1, each block B_i corresponding to a different bit F_i in F equal to 1 and having a size of 2^i.

88. (Original) The method of claim 86 further comprising:

performing a partial defragmentation on the storage medium if the request is not accepted; and

responsive to the performing step clearing a sufficient contiguous block on the storage medium, storing the data file in a block of the storage medium cleared by the performing step.

89. (Original) The method of claim 88 wherein the performing step comprises performing a heap manager partial defragmentation algorithm on the storage medium if the request is not accepted.

90. (Original) The method of claim 88 wherein the storage medium is a disk.

91. (Original) The method of claim 88 wherein the storage medium is computer memory.

92. (Currently Amended) A method of storing a data file on a storage medium, the data file having a file size comprising a total number of bytes therein, the method comprising:

maintaining a minimum size 2^m for a block of storage space into which the data file or a segment thereof will be stored;

determining whether the file size is an even power of 2;

if the file size is an even power of 2 and greater than or equal to 2^m, requesting a block of storage space on the storage medium equal to the file size;

if the file size is less than 2^m, requesting a block of storage space on the storage medium equal to 2^m;

if the file size is not an even power of 2 and greater than 2^m, requesting a plurality of blocks of storage space on the storage medium, each block having a size that is equal to a power of 2 and equal to or greater than 2^m; and

if the request is accepted, storing the data file on the storage medium in accordance with the request.

93. (Original) The method of claim 92 wherein the file size can be represented in binary as $F = F_k \ldots F_2F_1$, and wherein if the file size is not an even power of 2, the requesting step comprises:

if each bit F_i in $F_m \ldots F_1$ is equal to zero, then selecting a total number n of blocks $B_1, \ldots, B_n$ equal to a total number of bits in F equal to 1, each block B_i corresponding to a bit F_i in F equal to 1 and having a size of 2^i; and

if any bit F_i in $F_{m-1} \ldots F_1$ is equal to 1, then (1) rounding F up to a minimum value R that is greater than F for which each bit R_i in $R_{m-1} \ldots R_1$ equals zero, and (2) selecting a total number n of blocks $B_1, \ldots, B_n$ equal to a total number of bits in R equal to 1, each block B_i corresponding to a bit R_i in R equal to 1 and having a size of 2^i.

94. (Original) The method of claim 93 further comprising:

performing a partial defragmentation on the storage medium if the request is not accepted; and

responsive to the performing step clearing a sufficient contiguous block on the storage medium, storing the data file in a block of the storage medium cleared by the performing step.

95. (Original) The method of claim 94 wherein the performing step comprises performing a heap manager partial defragmentation algorithm on the storage medium if the request is not accepted.

96. (Original) The method of claim 95 wherein the storage medium is a disk.

97. (Original) The method of claim 95 wherein the storage medium is computer memory.

98. (New) The system of claim 40 wherein the programmable logic device is further configured to deactivate a stage of the plurality of stages, whereby the deactivated stage acts as a pass through for the data it receives.

99. (New) The system of claim 98 wherein the plurality of stages have an associated order, the order of the stages remaining the same whether any of the stages are deactivated.

100. (New) A data processing system comprising:

a data storage medium;

a processing device in communication with the data storage medium; and

a computer system having a system bus, wherein the computer system is configured to communicate with the processing device over the system bus;

wherein the processing device comprises a programmable logic device, the programmable logic device implementing a plurality of data processing stages;

wherein each data processing stage is dedicated to a different data processing operation;

wherein the data processing operations comprise at least two selected from the group consisting of a search operation, a data reduction operation, a data classification operation, an encryption operation, a decryption operation, a compression operation, and a decompression operation;

wherein the processing device is configured to selectively activate and deactivate individual ones of the data processing stages to define a data processing pipeline, wherein a deactivated stage acts as a pass through for the data it receives, and wherein an activated stage performs the data processing operation to which that stage is dedicated upon the data it receives; and

wherein the programmable logic device is further configured to process data, as it passes between the data storage medium and the computer system, through the defined data processing pipeline.

101. (New) The system of claim 100 wherein the programmable logic device comprises an FPGA.

102. (New) A hard disk drive accelerator for connection between a hard disk drive and a processor, said accelerator comprising reconfigurable hardware logic arranged such that data read from the hard disk drive streams through the reconfigurable hardware logic prior to being passed on to the processor, wherein the reconfigurable hardware logic is configured to process the data stream through a pipeline comprising a plurality of processing stages, each processing stage being configured to perform a data processing operation on the data it receives, wherein the processing operations performed by the stages of the pipeline are at least two selected from the group consisting of: a search operation, a data reduction operation, a data classification operation, an encryption operation, a decryption operation, a compression operation, and a decompression operation, and wherein the reconfigurable hardware logic is further configured to selectively deactivate a processing stage of the plurality of stages, whereby the deactivated stage acts as a pass through for the data it receives.

103. (New) The accelerator of claim 102 wherein the plurality of stages have an associated order within the pipeline, the order of the stages remaining the same whether any of the stages are deactivated.

104. (New) The accelerator of claim 103 wherein the reconfigurable hardware logic is further configured to selectively activate a deactivated stage of the pipeline, whereby the activated stage performs the processing operation to which that the stage is dedicated on the data that the activated stage receives.